IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 24, 2019, accepted December 10, 2019, date of publication December 20, 2019,
date of current version January 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961271

Improved Grid Algorithm Based on Star Pair Pattern and Two-dimensional Angular Distances for Full-Sky Star Identification

JIAN LI[1], XINGUO WEI[1], GANGYI WANG[1], AND SHUTIAN ZHOU[2]
[1]School of Instrument Science and Opto-electronics Engineering, Beihang University, Beijing 100191, China
[2]School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611730, China

Corresponding author: Gangyi Wang (wanggangyi@buaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61705005, Grant 61605007, and Grant 61378052, and in part by the Beijing Gold-Bridge Project under Grant ZZ19019.

ABSTRACT A fast and robust star identification algorithm is proposed in this paper. The algorithm is derived from the grid algorithm but significantly improves the robustness by increasing the probability of the correct close neighbor star and reducing the pattern noise. Firstly, star pair patterns are constructed rather than star patterns. And a traversal method is adopted to find all the possible closest neighbor stars, which improves the robustness towards brightness noise. Secondly, the two-dimensional angular distance features are used instead of grid features in Cartesian coordinates to improve the robustness towards positional noise. The simulation results show that the proposed algorithm is quite robust to a variety of noise conditions. The identification rate of the proposed algorithm is higher than the grid algorithm and similar to the pyramid algorithm, while the identification speed is up to dozens times faster than the pyramid algorithm.

INDEX TERMS Attitude sensor, star sensor, star identification.

NOMENCLATURE

$\boldsymbol{V}$ = visible stars
$\boldsymbol{C}$ = catalog stars
$\boldsymbol{S}$= sensor stars
c = catalog star
s = sensor star
λ = grid size
min_match = match threshold
$N_{\text{reference}}$ = number of stars to be identified
$N_{\text{pattern\_in\_cat}}$ = number of pattern stars in catalog
$N_{\text{neighbor\_in\_img}}$ = number of closest neighbor stars in image
$N_{\text{pattern\_in\_img}}$ = number of pattern stars in image

I. INTRODUCTION

Attitude measurement is the core technology for spacecraft flight and plays a key role in the development and performance improvement of a spacecraft. As a kind of high-precise attitude measurement device, star sensor is widely used in various spacecraft currently.

A star sensor uses stellar positions to generate attitude determinations. The difficulty in realizing attitude estimation lies in the full-sky star identification, i.e., finding the correspondence between measured and cataloged stars without a priori attitude knowledge input. This step extracts several rotation invariant features from a set of body vectors and associated brightness, searches the database to match the observations with the appropriate catalog stars, and performs validation to eliminate false star pairings. Several full-sky star identification algorithms have been proposed [1], but efficiency and reliability remain key problems.

Existing full-sky star identification algorithms can be partitioned into two classes. The first class of algorithms approaches star identification as an instance of sub-graph isomorphism. In this case, the stars are considered as vertices in a graph, the edges of which correspond to the angular distances between neighboring stars that could share the same field of view (FOV). Identification arises when the resultant graph obtained from the sensor image is uniquely identified

The associate editor coordinating the review of this manuscript and approving it for publication was Yonghong Peng.

with a portion of the database. Typical graphs include triangles [2] and quadrilaterals [3]. The triangle algorithm is easy to implement, but time-consuming due to the binary search and redundant matches verification. Through the use of the k-vector approach discussed in [4], [5], the pyramid algorithm solves the range-searching problem without a searching phase. The pyramid algorithm is robust to false stars, but the identification time increases exponentially with the number of false stars.

Some novel algorithms were presented to reduce identification time. A hash map of the triangle features and an optimized triangle feature selection strategy are introduced in [6], at the cost of the large database capacity which limits its on-orbit applications. A geometric voting algorithm proposed in [7] reduces the time spent in cross-checking the results of the k-vector range search but is susceptible to positional noise and false stars. Hence, the geometric voting algorithm is improved in [8] by adding an iterative process to mitigate the interference of false stars. However, the iterative process may not converge in case of fewer real stars.

Generally, the first class of algorithms is relatively mature and has been successfully applied in orbit. The expected frequency of matching an observed star polygon with an incorrect polygon in the star catalog due to measurement error were developed in [9] for the performance prediction and evaluation of sub-graph isomorphism algorithms. It is shown that the identification result is accurate enough in case of four real stars. Therefore, the pyramid algorithm is the most widely used algorithm in orbit.

The second class of algorithms tends to approach star identification in terms of pattern recognition. In this case, each star is associated with an individual pattern or signature determined by its surrounding star field and can thus be identified by finding the closest match in the catalog patterns. The grid algorithm is the most representative pattern-based approach, which was first discussed in [10] and further refined by other researchers [11]–[13]. These algorithms provide robust star identification over a wide range of sensor noise for small FOV sensors with high star magnitude sensitivity because more stars could be observed. However, it is difficult to apply the algorithms in the most current systems due to the low probability of finding the correct close neighbor star and less observed stars.

Some techniques are employed to avoid the use of unreliable information on the close neighbor star. Radial and cyclic features were adopted, and a multistep match routine was applied in [14]. This identification strategy requires additional checking to preserve integrity. A log-polar transforming-based star pattern was introduced in [15], while a star pattern called flower code was presented in [16]. These patterns are less efficient than others because of their cyclic dynamic match step.

Overall, the second class of algorithms is more efficient and inherently more robust due to the use of high-dimensional features for identification. Although it has not yet been applied successfully in orbit, there is still much room for improvement.

In recent years, some new star identification algorithms were developed. For instances, the shortest distance transform based algorithm [17], Euclidean distance transform based algorithm [18], and the multi-poles algorithm [19] utilize the features composed of multiple vertices so that the matching process is more complicated. Although the robustness is high, the identification time is long. The polygon algorithm [20] and the triangle voting algorithm [21] reduce the feature dimension which improve the efficiency but reduce the robustness. These algorithms cannot balance the efficiency and the robustness and be directly applied in actual star sensor.

In this paper, an improved grid algorithm is proposed to meet the requirements of large FOV and low sensitivity. Firstly, star pair patterns are constructed rather than star patterns. Secondly, the two-dimensional angular distance features are used instead of grid features in Cartesian coordinates. Compared with the original grid algorithm, the proposed algorithm effectively increases the probability of finding the correct close neighbor star and reduces the pattern noise. Therefore, the robustness is improved while the efficiency is maintained so that the proposed algorithm is especially suitable for on-orbit applications.

The remainder of this paper is organized as follows. A brief introduction on grid method is given in Section II. Section III describes the algorithm, including the pattern construction, database generation, and matching criteria, in detail. Section IV presents the simulations to determine the parameter values and demonstrates the performance of the proposed algorithm. Finally, some conclusions are drawn.

II. GRID ALGORITHM

As shown in Figure 1, a grid pattern is generated in the following manner. For a reference star r, a pattern area is defined as $sky(r, pr)$ which includes all the stars falling inside a circle region of radius pr around r. The closest star to r inside $sky(r, pr)$ and outside a buffer radius br is considered as its closest neighbor star denoted by l. Thereafter, $sky(r, pr)$ is shifted until r is located on the center of FOV. A square grid of size $g \times g$ is superposed on pattern area such that its center is located on r and its horizontal axes lies on the straight line passing through the reference star and its closest neighbor star. Finally, the pattern is derived so that each grid cell that contains at last one star is labeled by 1 and the empty cells are labeled by 0.

To determine which pattern in the database is associated with a particular sensor pattern, the sensor pattern is compared with the patterns in the database to find the best match that has the greatest number of non-zero cells in common. If the number of shared cells between the best catalog pattern and the sensor pattern is greater than some threshold, the reference star of the catalog pattern is paired with the reference star of the sensor pattern.

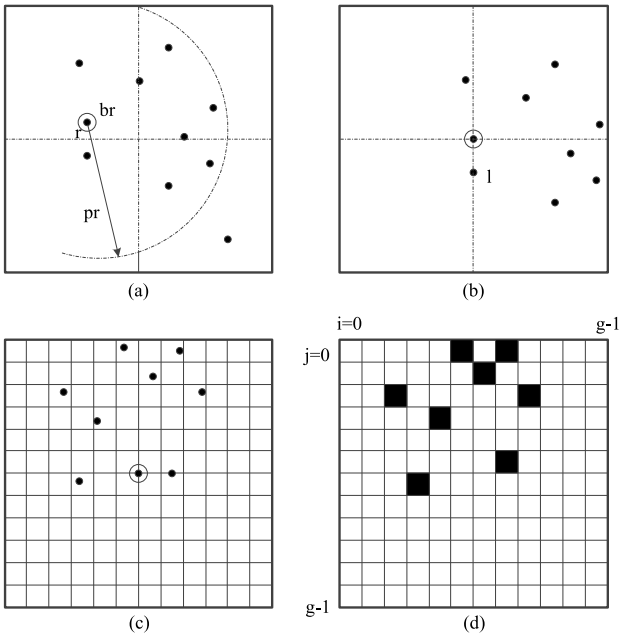


FIGURE 1. Grid pattern generation. (a) Pattern area is defined as $sky(r, pr)$ which includes all the stars falling inside a circle region of radius pr around the reference star r. (b) The pattern area is shifted until r is located on the center of FOV. (c) A square grid of size $g \times g$ is superposed on pattern area such that its center is located on r and its horizontal axes lies on the straight line passing through the closest neighbor star. (d) All stars except reference star are projected down on grid so that the shaded grid cells constitute the star pattern.

There are some shortcomings in feature extraction for the grid algorithm mainly in two aspects. Firstly, the probability of correctly selecting the closest neighbor star is low such that it is about 50% even at no positional noise. The loss of the correct closest neighbor star can occur in a number of ways. On one hand, the reference star may be too close to the edge of the FOV, and its closest neighbor star is out of the FOV. On the other hand, the closest neighbor star could possibly be near the sensitivity limit of the sensor, so that it may not appear in the image due to noise. A related phenomenon is the addition of a star below the limit of the sensor that satisfies the criteria for being the closest neighbor. In the case of the wrong closest neighbor star, an erroneous star pattern will be generated from the sensor image and not be matched against the database pattern.

Secondly, the star pattern is susceptible to position noise. As shown in Figure 2 (a), since the distance from the reference star to the closest neighbor star is relatively closer, the pattern noise of the neighbor stars further from the reference star is amplified, which may result in an incorrect star pattern. The larger the FOV, the larger the pattern noise, and the larger the possibility of erroneous star pattern. For an instance shown in Figure 2 (b), the true star pattern (marked by black dots) is {16, 19, 39, 45, 51}. Due to position noise, the observed star pattern (marked by white dots) is {11, 24, 39, 44, 51}. It can be seen that star pattern obtained from similar star distribution will not be correctly matched.

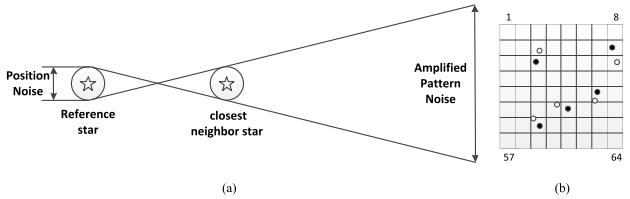


FIGURE 2. Amplification effect of position noise on the generation of star pattern. (a) The pattern noise of the neighbor stars further from the reference star is amplified due to position noise. (b) The correct star pattern marked by black dots and the erroneous star pattern by white dots.

The grid algorithm provides robust star identification over a wide range of sensor noise for small FOV sensors with high star magnitude sensitivity because more stars could be observed. In the case that at least two star are identified, the attitude could be determined. Specifically, an 8×8 degree FOV and a guide star catalog with apparent magnitudes ranging down to 7.5 are used in the simulations so that the average number of stars in the FOV is about 40. Therefore, even if some sensor star patterns are incorrect leading to misidentification, there are adequate correctly identified stars. While the number of average stars in the field of view decreases, the identification rate of the grid algorithm is significantly reduced.

In this paper, the grid algorithm is improved to avoid the above disadvantages. Firstly, the method of selecting the closest neighbor star is improved. Considering that the darker stars are greatly affected by noise, the brighter stars are used as neighbor stars. At the same time, the traversal method is adopted to find all the possible closest neighbor stars. And each star pair of the reference star and the closest neighbor star is stored. Secondly, the angular distances of the other stars to the reference star and the closest neighbor star are used to generate the patterns. Thus, the pattern elements are of equal accuracy, and the pattern noise of the distant stars is reduced. As for the pattern match, the similar pattern elements are found instead of the same pattern elements. Based on the above improvements, the robustness is improved while the efficiency is maintained.

III. IMPROVED GRID ALGORITHM

In this section we initially specify a description of the star pair pattern and the matching criteria. For practical reasons, we also provide a number of implementation details regarding database generation and data structures suitable for efficient matching. The algorithm implementation is then presented.

Visible stars ($\boldsymbol{V}$). Stars that can actually be imaged by the sensor.

Catalog stars ($\boldsymbol{C}$). A subset of $\boldsymbol{V}$ contained in the database used on board the spacecraft. The set stars are defined in the inertial reference frame and used to estimate the attitude.

Sensor stars ($\boldsymbol{S}$). Set of detected objects on the sensor focal plane, which includes a subset of $\boldsymbol{V}$ and spurious non-star items such as sensor noise.

To facilitate analysis and understanding of the implementation details, some terms are introduced for stars in different reference frames or those that have special uses.

The relation of inclusion among $\boldsymbol{V}$, $\boldsymbol{C}$, and $\boldsymbol{S}$ is presented in Figure 3. Star identification determines the correspondence between the sensor and catalog stars in the shaded region. We typically use c when referring to catalog stars and s when the star is from the sensor image.

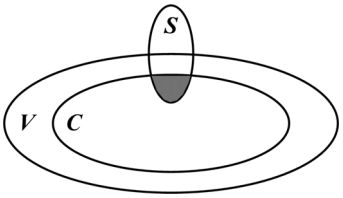


FIGURE 3. Relationship between different star sets.

A. PATTERN GENERATION AND MATCHING

The patterns are constructed in the following manner as shown in Figure 4.

1) Find all the possible closest neighbor star for each reference star. Then the pattern for each star pair of the reference star and the closest neighbor star is generated.
2) Find the nearest N_{pattern} stars around the star pair as pattern stars. Calculate the angular distances from each pattern star to the reference star and the closest neighbor star respectively, denoted by α and β.
3) Divide the two-dimensional space composed of α and β into a $g \times g$ grid. In this way, the pattern is derived as a g-order square matrix so that each cell that contains at last one pattern star is 1 on the corresponding element and those without are 0.

Given a pattern in the sensor image pat^s and a set of patterns $\left\{pat_i^c | i = 0, 1, \ldots, N_D\right\}$ from the database, the candidate match of the sensor star pairs in the database should satisfy

$$\max_i \sum_{a=1}^{g}\sum_{b=1}^{g}\sum_{m=a-1}^{a+1}\sum_{n=b-1}^{b+1}\left(pat^s(a,b)\ \&\ pat_i^c(m,n)\right) \geq min\_match \tag{1}$$

Where N_D is the number of the patterns in the database, and & is the logical AND operation. If there are non-zero elements in adjacent areas, the two pattern elements in the database and sensor image are considered matched. This is because the stars at the edge of the grid are easily affected by noise and fall into the adjacent grids. The match threshold *min_match* determines the error rate of the algorithm. The relationship between them has been analyzed thoroughly in [9]. The greater *min_match* is, the smaller the error rate is. It is set to 3 for an extremely low error rate, as a false attitude is unacceptable for practical applications.

B. DATABASE GENERATION

The proposed algorithm uses a database that consists of a catalog of known star locations in the celestial coordinate system, a list of star pair distances, and the patterns for the star pairs.

1) STAR CATALOG

Generally, some of the stars in $\boldsymbol{V}$ are unsuitable for navigation purposes. These stars are binary stars or have variable brightness that causes difficulties during identification. More importantly, the distribution of stars in $\boldsymbol{V}$ varies significantly over the sky [22]. Including all the suitable stars in $\boldsymbol{C}$ could thus bias the identification routine and seriously degrade its performance.

Several methods could be used to determine which stars should be included in $\boldsymbol{C}$. For our purpose, a relatively simple procedure is sufficient. After discarding binary stars and variable stars, an incremental, uniform scan is performed across the celestial sphere and the brightest $N_{\text{reference}}$ stars within the circular FOV are added at each orientation if they are not elements of $\boldsymbol{C}$. Some parts of the sky could have more stars than $N_{\text{reference}}$ because no attempt was made to remove possibly redundant stars. The $N_{\text{reference}}$ value is influenced by the expected noise level incorporated in the image. More stars in $\boldsymbol{C}$ for a section of the sky increase the identification rate. For our algorithm simulation and testing, we set $N_{\text{reference}}$ to 15 so that the star catalog consists of 4285 stars.

2) STAR PAIR LIST

The star distribution in the neighborhood of the reference star is related to its position in the FOV. There are different closest neighbor stars for the reference star at different positions. For a complete star pair list, all the possible closest neighbor stars are found by traversal search to pair with the reference star.

The star tracker optical system is usually aligned in a defocused position so that the size of the star image is 3×3 to 5×5 pixels to improve the star location measurement accuracy [23]. Hence, the star tracker needs rotate $3 \times \text{FOV}/N_{\text{pixel}}$ to $5 \times \text{FOV}/N_{\text{pixel}}$ perpendicular to the boresight to ensure that a star completely enters the FOV, where N_{pixel} is pixel resolution of the image sensor. The traversal search could be performed as follows.

Firstly, a circle region of radius FOV/2 around the reference star in which the reference star could be imaged is equally divided into $N_{\text{boresight}}$ regions in two directions perpendicular to each other with the interval equal to $3 \times \text{FOV}/N_{\text{pixel}}$. Secondly, the brightest $N_{\text{reference}}$ stars within the FOV are found for each boresight of each region center. If the reference star is one of the brightest $N_{\text{reference}}$ stars, then the star pair of the reference star and the closest star in the brightest $N_{\text{reference}}$ stars is added. For our algorithm simulation and testing, we set FOV to 16 degree so that the list consists of 13588 star pairs.

For an efficient matching operation and less memory consumption, all the star pairs are sorted in ascending order and

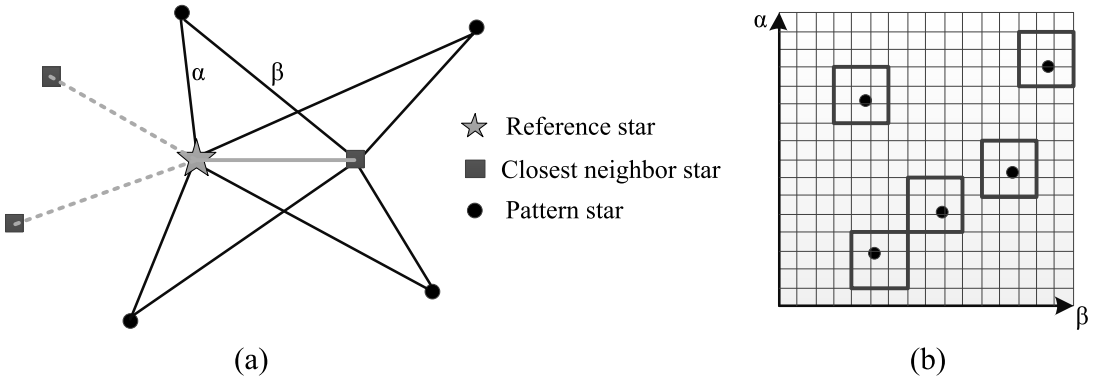


FIGURE 4. Improved grid algorithm. (a) Pattern Generation. (b) Pattern Matching. The bold grid means that the two pattern elements in the database and sensor image are considered matched if there are non-zero elements in adjacent areas.

equally divided into a number of groups with the size λ in terms of separation angle [24]. Thus, the list entries of the sensor distance θ are located by the following equations.

$$m_{start} = INT(\frac{\theta}{\lambda}) - 1$$
$$m_{end} = INT(\frac{\theta}{\lambda}) + 1 \quad (2)$$

INT (a) indicates the largest integer that is not larger than a.

The star pairs, including the groups from m_{start} to m_{end}, are considered matched. This is because the star pairs at the edge are easily affected by positional noise and fall into the adjacent groups.

3) PATTERN DATABASE

For each pair in the database, the nearest N_{pattern} catalog stars around its geometric center are found as pattern stars. Then each pattern star is arranged in a clockwise direction with the two vertices of the star pair as follows: the pattern star, vertex 1 and vertex 2. The angular distances from the pattern star to vertex 1 and vertex 2 are calculated respectively, denoted by α and β, and quantized according to the grid size λ. Finally, the pattern of the star pair is derived as

$$\left\{ (INT\ (\alpha_i/\lambda)\ , INT\ (\beta_i/\lambda)) \,\middle|\, i = 1, 2, \ldots, N_{\text{pattern}} \right\} \quad (3)$$

The structure of the patterns as described in equation (3) is unsuitable for an efficient matching operation and consumes more memory than needed. Hence, the patterns are incorporated into a hash table H. We construct a function called hash function to map each pattern element to an integer which serves as the table index. Each entry in the table contains all the star pairs whose patterns contain the corresponding element. Limited by on-orbit memory capacity, we set $N_{\text{pattern\_in\_cat}}$ to 20 in simulation so that the hash table consists of 271760 star pairs. To find and calculate the maximum match for a sensor pattern, we first bring each element index into the function, get the table index, then examine the hash table at each index and increase the counter for each star pair listed in the database. Figure 5 demonstrates the hash method.

Assumed that M is the number of table entries mapped by pattern elements, and then we need to find a hash function f satisfying

$$f(a_i, b_i) = H_i,$$
$$\text{where } H_i \in [1, M], \text{ and } \forall i \neq j, H_i \neq H_j \quad (4)$$

There are already some effective algorithms for constructing such functions. We adopt the algorithm in [25] as it is fast and memory saving.

C. ALGORITHM IMPLMENTATION

The algorithm begins with input of the measurement of the positions and the apparent brightness of the sensor stars. Testing all of the sensor stars can increase the risk of generating spurious matches and degrade the time and memory performance. Thus, only the $N_{\text{reference}}$ brightest sensor stars are identified in ascending order of distance to the center of the FOV as the pattern near the center of the FOV is more comprehensive. The identification is divided into two steps: star pair matching and pattern matching.

1) STAR PAIR MATCHING PROCESS

Considering that each of the $N_{\text{reference}}$ sensor stars is a reference star, the $N_{\text{neighbor\_in\_img}}$ closest neighbor stars to the reference star are found, as the matching star pair of the reference star and the closest neighbor star may not be included in the star pair database due to noise. According to the simulation result in Section IV-A, $N_{\text{neighbor\_in\_img}}$ is set to 2.

During the implementation, a set of counters is assigned to each sensor star pair. And each counter is associated with each star pair in the angle distance database. The angle distance of the sensor star pair θ is calculated and used to find the groups in the star pair database within $(INT(\theta/\lambda) - 1, INT(\theta/\lambda) + 1)$, where λ denotes the group size. For each star pair in the database that matches with a sensor pair, the corresponding counters are set to 1 while the others are set to 0.

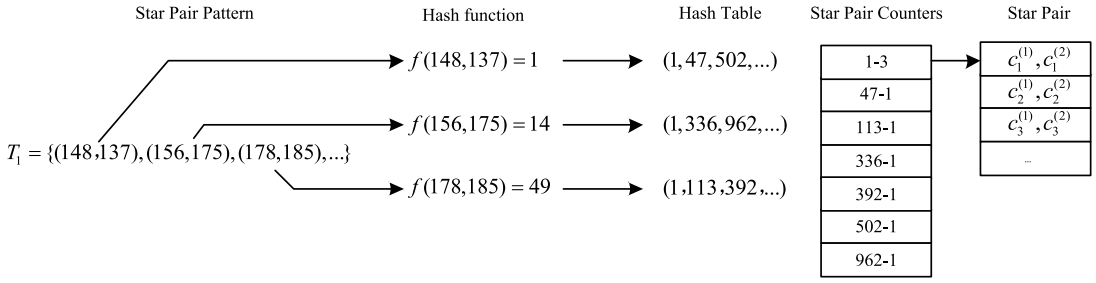


FIGURE 5. Illustration of the star pair pattern storage and matching.

2) PATTERN MATCHING PROCESS

For the sensor pair mentioned above, the $N_{\text{pattern\_in\_img}}$ closest stars around its geometric center are found as pattern stars. Then the angular distances from each pattern star to the vertices of the star pair are calculated, quantized, and brought into the hash function to get the hash table index. The counter for each star pair in each entry of the hash table is examined. If its value is greater than 0, the counter increases by 1. At the end of this procedure, the unique counter with the highest value is the pattern in the database closest to the sensor pattern. If the counter value is greater than the match threshold *min_match*, the star pair in the database is paired with the sensor star pair and the algorithm reports a success.

It is noted that the pattern match should be performed in clockwise and counterclockwise directions separately. If clockwise identification is successful, the vertex order of the sensor star pair is consistent with the cataloge star pair; otherwise, the vertex order of the sensor star pair is opposite to the cataloge star pair.

IV. SIMULATION RESULTS

Several simulations were performed to evaluate the proposed algorithm under a variety of noise conditions and compare it with the pyramid and grid algorithms which are the most representative subgraph-based and pattern-based approach respectively. These algorithms were implemented in MATLAB in the Microsoft Windows environment on a Core i3 2.5 GHz PC.

We developed a platform used in star identification testing to simulate the picture taken by the star sensor for the simulation. It allows testing with an arbitrary star catalog for a number of user-specified sensor configurations and noise levels. A guide star catalog is made up of 25848 stars with apparent magnitudes ranging down to 7.5 based on the J2000 catalog.

The star sensor configuration used for the simulations reported in this paper uses a 16 degrees circular FOV with a focal plane consisting of 1024 $\times$ 1024 pixels so that each pixel subtends a square area of approximately 56.25 arc seconds. The minimum sensitivity of the sensor was set to 6 Mv apparent stellar magnitude.

Noise terms are also included in the sensor configuration. Three sources of noise are introduced in the simulation, namely, brightness and location uncertainty, and false stars. The noise addition to the brightness of the imaged stars allowed dimmer stars to actually appear during the simulation and for brighter stars to be lost. Aside from the changes made to the observed brightness for each star, random Gaussian noise is also added to the sensor star locations in terms of pixels. This results in the deviation from the correct patterns in the database. Besides the measurement noise, spurious non-star items may appear in the image. The algorithm's robustness to the noise is examined thoroughly. The algorithm parameters are fixed and a single noise source is increased, whereas the others are maintained at typical levels in the simulation. The statistics reported in this section include 10000 random sensor orientations that are uniformly distributed over the celestial sphere.

The following results are presented. First, the appropriate parameter values are determined, including the group size or grid size (λ), the number of closest neighbor stars ($N_{\text{neighbor\_in\_img}}$) and the number of pattern stars ($N_{\text{pattern\_in\_img}}$). The identification rate was then measured under a variety of different noise conditions. The memory and time performance was also described.

A. PARAMETERS SELECTION

1) SELECTION OF THE GROUP SIZE OR GRID SIZE (λ)

As the common parameter of the proposed algorithm and pyramid algorithm, the group size λ has an important impact on the algorithm robustness. The parameter should be increased with a higher location error of the observed stars on the sensor focal plane for a higher identification rate. However, the greater group size would result in more spurious matches and consequently greater match threshold *min_match*. For the simulation, we set λ to 0.04 degree, which equals 2.6 pixels approximately in the image plane.

2) SELECTION OF THE NUMBER OF CLOSEST NEIGHBOR STARS ($N_{\text{neighbor\_in\_img}}$)

Although the method of selecting neighboring stars is improved in the database generation, the sensor star pair may not be included in the star pair database due to noise.

TABLE 1. Average number of neighbor stars with different positional noise.

Positional noise (pixel)	0	0.4	0.8	1.2	1.6	2.0
Average number of neighbor stars	1.24	1.24	1.25	1.26	1.28	1.29

TABLE 2. Average number of neighbor stars with different brightness noise.

Brightness noise (Mv)	0	0.2	0.4	0.6	0.8	1.0
Average number of neighbor stars	1.09	1.18	1.26	1.34	1.38	1.39

TABLE 3. Average number of neighbor stars with different number of false stars.

Number of false stars	0	1	2	3	4	5
Average number of neighbor stars	1.25	1.33	1.37	1.40	1.41	1.42

Tables 1 to 3 show the changes in the average number of neighbor stars required to find the correct match in the database for the reference star along with the increase of the positional noise, brightness noise and the number of false stars respectively. It can be seen that the average number of neighbor stars is little affected by positional noise and not greater than 2 for brightness noise and false stars. As the common parameter of the proposed algorithm and grid algorithm, we set $N_{\text{neighbor\_in\_img}}$ to 2 in the simulation.

3) SELECTION OF THE NUMBER OF PATTERN STARS ($N_{\text{pattern\_in\_img}}$)

Some sensor patterns are not included in the database due to noise. Tables 4 to 6 show the changes in the average number of pattern stars required to find the correct match in the database for the sensor star pair along with the increase of the positional noise, brightness noise and the number of false stars respectively. It can be seen that the average number of pattern stars is little affected by positional noise and not greater than 9 for brightness noise and false stars. As the specific parameter of the proposed algorithm, we set $N_{\text{pattern\_in\_img}}$ to 10 in the simulation.

B. ROBUSTNESS TOWARDS DIFFERENT NOISE

1) ROBUSTNESS TOWARDS POSITIONAL NOISE

Figure 6 shows the identification rate for the proposed algorithm, pyramid algorithm, and grid algorithm as the amount

TABLE 4. Average number of pattern stars with different positional noise.

Positional noise (pixel)	0	0.4	0.8	1.2	1.6	2.0
Average number of pattern stars	5.75	5.93	6.04	6.14	6.21	6.35

TABLE 5. Average number of pattern stars with different brightness noise.

Brightness noise (Mv)	0	0.2	0.4	0.6	0.8	1.0
Average number of pattern stars	4.09	4.91	5.96	7.23	8.19	8.72

TABLE 6. Average number of pattern stars with different number of false stars.

Number of false stars	0	1	2	3	4	5
Average number of pattern stars	5.86	6.68	7.33	7.79	8.17	8.58

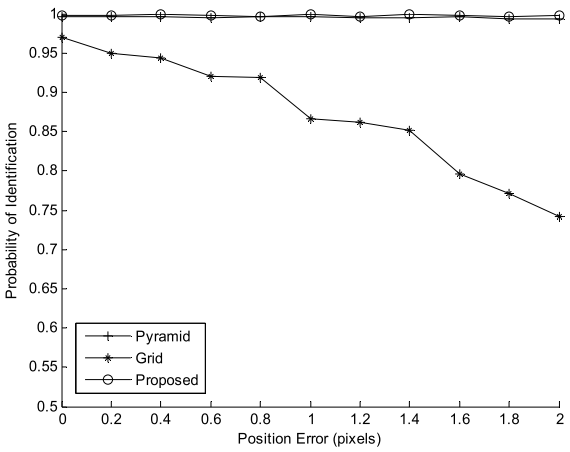


FIGURE 6. Identification rate versus position accuracy.

of positional noise added to the image increases from 0.0 to 2.0 pixels. The brightness deviation was maintained at 0.4 Mv stellar magnitude resulting in the loss and addition of visible stars. Nearly 5.9 unknown stars below the minimum sensitivity could be imaged, and approximately 3 expected stars were eliminated.

The proposed algorithm appears to be quite stable whose identification rate is maintained at 99.8% as the standard deviation of star location error increases from 0 to 2 pixels. This condition is almost the same as the pyramid algorithm and better than the grid algorithm, especially for higher noise levels. At a 2-pixel deviation, the identification rate of our

algorithm is 25% higher than the grid algorithm. In summary, the proposed algorithm and pyramid algorithm have higher robustness to positional noise.

2) ROBUSTNESS TOWARDS BRIGHTNESS NOISE

Figure 7 shows the identification rate for the 3 algorithms as brightness accuracy decreases with the location deviation at 0.5 pixels. Due to unexpected additions and deletions of stars in the observed sky image, the identification rate of the proposed algorithm drops from 99.8% to 93.8%, while the pyramid algorithm hardly decreases. Approximately 6.0 stars per image disappeared on average for a 1.0 Mv standard deviation in stellar magnitude. Approximately 16.6 extra stars that would not normally be seen were also imaged at 1.0 Mv deviation. Among the 15 brightest stars to be identified, 5.4 extra stars appeared on average at 1.0 Mv. The condition for a successful identification is to find the correct correspondence of 4 image stars for the pyramid algorithm, and 5 image stars for the proposed algorithm. The probability of containing 5 true stars among the identified stars is less than the probability of containing 4 true stars so that the identification rate of the proposed algorithm is less than the pyramid algorithm at higher brightness noise.

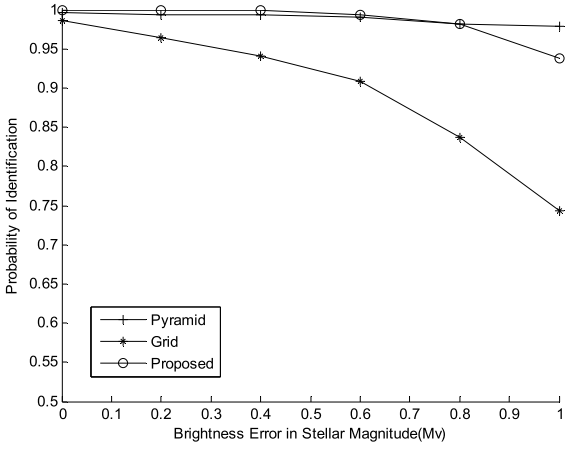


FIGURE 7. Identification rate versus brightness accuracy.

The identification rate of our algorithm is 20% higher than the grid algorithm and 4% lower than the pyramid algorithm at 1.0 Mv deviation. The results show that the robustness of the proposed algorithm is better than the grid algorithm, but slightly worse than the pyramid algorithm.

3) ROBUSTNESS TOWARDS FALSE STARS

Figure 8 shows the identification rate with respect to the number of false stars. The acquired images can contain false stars because of hot spots, radiation, planets, or observed spacecraft. The influence of false stars is investigated as the number of false stars injected into the 15 brightest stars to be identified at random locations increases from 1 to 5. The brightness deviation was maintained at 0.4 Mv and location deviation at 0.5 pixels for the experiment. It can be seen that

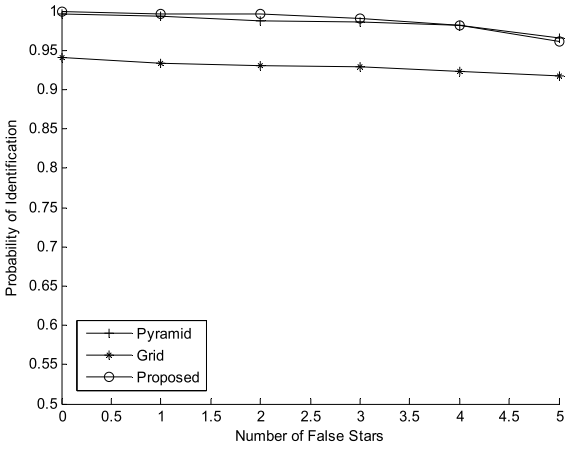


FIGURE 8. Identification rate versus number of false stars.

all the 3 algorithms have high robustness towards false stars so that the identification rate does not decrease significantly with the increase of the number of false stars.

The proposed algorithm and pyramid algorithm identified stars at a rate higher than 98% up to a noise level of 4 false stars and 96% in case of 5 false stars. Due to the position and brightness noise, the identification rate of the proposed algorithm is 5% higher than the grid algorithm.

4) ROBUSTNESS TOWARDS SENSITIVITY REDUCTION

Figure 9 shows the identification rate for the 3 algorithms as sensitivity decreases with the brightness deviation at 0.4 Mv and the location deviation at 0.5 pixels. The sensitivity reduction results in the loss of visible stars and the average number of the expected stars which were eliminated increases from 3.3 to 9.4 as the sensitivity deceases from 0 to 0.5 Mv. It is seen that the proposed algorithm appears to be quite robust to sensitivity reduction and its identification rate is greater than 97.8%. This condition is almost the same as the pyramid algorithm and better than the grid algorithm, especially for lower sensitivity. At a 0.5 Mv reduction,

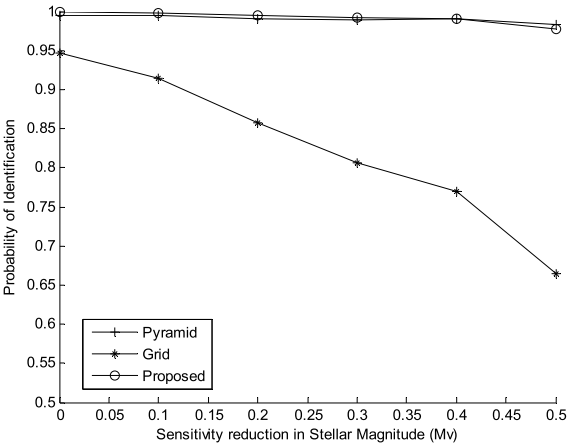


FIGURE 9. Identification rate versus sensitivity reduction.

the identification rate of our algorithm is 31% higher than the grid algorithm. In summary, the proposed algorithm and pyramid algorithm have higher robustness to sensitivity reduction.

C. MEMORY AND TIME PERFORMANCE

The amount of memory consumed by the database for the proposed algorithm is easy to calculate. Firstly, the guide star catalog consists of 4285 stars, and each star requires 2 floats to specify locations in the celestial coordinate system. The memory consumed by the guide star catalog is 0.03 Mb. Secondly, there are 13588 star pairs in the angular distance database, and each pair requires 4 bytes to store the indexes of the 2 catalog stars. The memory consumed by the star pair database is 0.05 Mb. thirdly, there are 20 elements per star pair pattern, and each element requires 2 bytes to store the associated pair index. The memory consumed by the pattern database is 0.52 Mb. The total memory usage of the database is 0.6 Mb. Similarly, the total memory usage for the grid algorithm and pyramid algorithm is 0.29 Mb and 0.83 Mb respectively. The consumed memory of the proposed algorithm is at a medium level to meet the requirements for on-orbit use.

TABLE 7. Identificaiton time with differnent positional noise (ms).

Positional noise (pixel)	0	0.4	0.8	1.2	1.6	2.0
Pyramid	125.8	103.2	136.6	162.6	206.2	372.0
Grid	4.7	5.4	5.8	6.5	7.4	8.8
Proposed	6.5	6.0	6.0	6.1	6.3	6.6

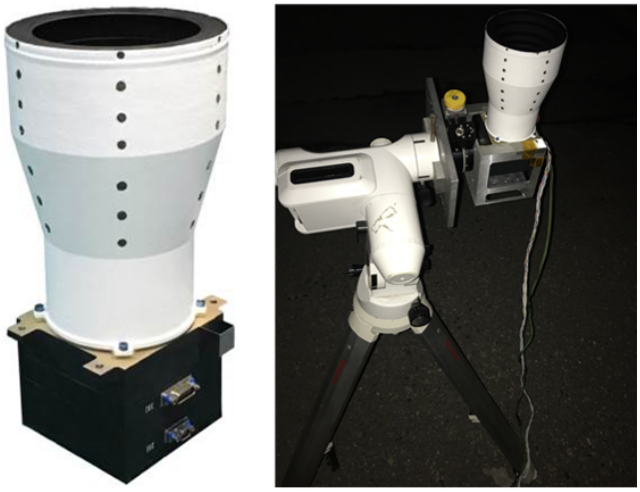
TABLE 8. Identificaiton time with differnent brightness noise (ms).

Brightness noise (Mv)	0	0.2	0.4	0.6	0.8	1.0
Pyramid	103.0	101.2	117.4	193.8	416.3	460.1
Grid	4.1	4.3	5.0	6.3	8.2	11.0
Proposed	4.1	4.5	6.2	9.5	13.1	19.2

Tables 7 to 9 show how the identification time changes for the 3 algorithms in a variety of different noise conditions. The identification time for the pyramid algorithm is very sensitive to sensor noise. The number of spurious matches increases with the noise so that it would take more time to determine the correspondence. The run time is 117 ms with a location error of 1.0 pixel and brightness error of 0.4 Mv stellar magnitude. The identification time for the grid algorithm is very sensitive to the positional and brightness noise, while the proposed algorithm is very sensitive to the brightness noise and false stars. The average run time of the proposed algorithm and grid algorithm is one order of magnitude smaller than the pyramid algorithm.

TABLE 9. Identificaiton time with differnent number of false stars (ms).

Number of false stars	0	1	2	3	4	5
Pyramid	117.4	181.1	203.4	245.3	339.0	477.1
Grid	5.0	5.7	5.8	6.2	6.6	6.9
Proposed	6.9	7.7	8.6	11.1	13.8	19.9



(a) (b)

FIGURE 10. Star sensor used for the night sky experiments. (a) Our own star sensor. (b) Night sky experiments.

D. COMPARISON AND ANALYSIS

The simulation results show that our algorithm is more robust than the grid algorithm at comparable memory and time consumption. Firstly, the traversal method of selecting the closest neighbor star improves the probability of selecting correct neighboring stars resulting in better robustness towards brightness noise. Secondly, the two-dimensional angular distance features are used to improve the robustness towards positional noise.

The proposed algorithm runs faster than the pyramid algorithm at comparable identification rate. The reasons are as follows: given $N_{\text{reference}}$ sensor stars, two patterns are established for each star and a total of $2N_{\text{reference}}$ patterns are recognized for the proposed algorithm, while a total of $C^3_{N_{\text{reference}}} = \frac{N_{\text{reference}}(N_{\text{reference}}-1)(N_{\text{reference}}-2)}{6}$ triangles are recognized for the pyramid algorithm. Hence, in the aspect of time performance, The computational complexity of the proposed algorithm is only $O(N_{\text{reference}})$, while the pyramid algorithm is $O(N^3_{\text{reference}})$.

E. NIGHT SKY EXPERIMENTS

We also tested the algorithm on real images besides the simulation. We developed a CMOS APS star sensor, which is shown in Fig. 9(a). It makes use of a 17° circular FOV with a focal plane consisting of 2048 × 2048 pixels. The night sky tests have been performed at Xinglong Astronomical Observatory of Chinese Academy of Sciences. The star sensor was

placed on an equatorial mount as shown in Figure 10 (b). We did not perform any algorithm parameter readjustment for this specific sensor. There were 20∼30 stars in the FOV on average, and all position error of the observed stars were below 2 pixels. The proposed algorithm successfully identified all of the 1500 images of random sensor orientations taken on several nights.

V. CONCLUSION

In this paper an improved grid algorithm for autonomous star identification is proposed. The algorithm has the following characteristics: 1) The traversal method is adopted to find all the possible closest neighbor stars, improving the robustness towards brightness noise. 2) The angular distances to the reference star and the closest neighbor star are used to generate the patterns, improving the robustness towards positional noise.

The simulation results demonstrate that the proposed algorithm provides robust star identification over a wide range of sensor noise, including positional and brightness error, as well as false stars. The identification rate maintains higher than 98% even if the standard deviation of position noise is up to 2.0 pixels, or the brightness noise is up to 0.8 Mv, or the number of false stars is up to 4, which is significantly higher than the grid algorithm and is similar to the pyramid algorithm, but the proposed algorithm is up to dozens times faster than the pyramid algorithm. Future work can involve incorporating the algorithm into a real star sensor for on-orbit validation.

REFERENCES

[1] B. B. Spratling and D. Mortari, "A survey on star identification algorithms," *Algorithms*, vol. 2, no. 1, pp. 93–107, Jan. 2009.

[2] C. C. Liebe, "Pattern recognition of star constellations for spacecraft applications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 8, no. 1, pp. 31–39, Jan. 1993.

[3] D. Mortari, M. A. Samaan, C. Bruccoleri, and J. L. Junkins, "The pyramid star identification technique," *Navigation*, vol. 51, no. 3, pp. 171–183, Aug. 2004.

[4] D. Mortari, "Search-less algorithm for star pattern recognition," *J. Astron. Sci.*, vol. 45, no. 2, pp. 179–194, Apr. 1997.

[5] D. Mortari, "K-vector range searching techniques," *Adv. Astron. Sci.*, vol. 105, no. 1, pp. 449–464, Jan. 2000.

[6] G. Wang, J. Li, and X. Wei, "Star identification based on hash map," *IEEE Sensors J.*, vol. 18, no. 4, pp. 1591–1599, Nov. 2018.

[7] M. Kolomenkin, S. Pollak, I. Shimshoni, and M. Lindenbaum, "Geometric voting algorithm for star trackers," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, pp. 441–456, Apr. 2008.

[8] J. Li, X. Wei, and G. Zhang, "Iterative algorithm for autonomous star identification," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 536–547, Jan. 2015.

[9] M. Kumar and D. Mortari, "An analytical approach to star identification reliability," *Acta Astronautica*, vol. 66, no. 3, pp. 508–515, Feb. 2010.

[10] C. Padgett and K. Kreutz-Delgado, "A grid algorithm for autonomous star identification," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 1, pp. 202–213, Jan. 1997.

[11] H. Lee and H. Bang, "Star pattern identification technique by modified grid algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 1112–1116, Jul. 2007.

[12] M. Na, D. Zheng, and P. Jia, "Modified grid algorithm for noisy all-sky autonomous star identification," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 2, pp. 516–522, Apr. 2009.

[13] M. Aghaei and H. A. Moghaddam, "Grid star identification improvement using optimization approaches," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 5, pp. 2080–2090, Oct. 2016.

[14] G. Zhang, X. Wei, and J. Jiang, "Full-sky autonomous star identification based on radial and cyclic features of star pattern," *Image Vis. Comput.*, vol. 26, no. 7, pp. 891–897, Jul. 2008.

[15] X. Wei, G. Zhang, and J. Jiang, "Star identification algorithm based on log-polar transform," *AIAA J. Aerosp. Comput., Inf. Commun.*, vol. 6, no. 8, pp. 483–490, Aug. 2009.

[16] J. Gong, L. Wu, J. Gong, J. Ma, and J. Tian, "Flower algorithm for star pattern recognition in space surveillance with star trackers," *Opt. Eng.*, vol. 48, no. 12, pp. 124401–124408, Dec. 2009.

[17] T. Delabie, T. Durt, and J. Vandersteen, "Highly robust lost-in-space algorithm based on the shortest distance transform," *J. Guid., Control, Dyn.*, vol. 36, no. 2, pp. 476–484, Jan. 2013.

[18] J. Roshanian, S. Yazdani, and M. Ebrahimi, "Star identification based on Euclidean distance transform, Voronoi tessellation, and k-nearest neighbor classification," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 6, pp. 2940–2949, Dec. 2016.

[19] V. Schiattarella, D. Spiller, and F. Curti, "A novel star identification technique robust to high presence of false objects: The multi-poles algorithm," *Adv. Space Res.*, vol. 59, no. 8, pp. 2133–2147, Apr. 2017.

[20] E. A. Hernández, M. A. Alonso, E. Chávez, D. H. Covarrubias, and R. Conte, "Robust polygon recognition method with similarity invariants applied to star identification," *Adv. Space Res.*, vol. 59, no. 4, pp. 1095–1111, Feb. 2017.

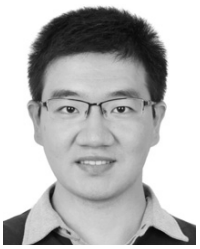
[21] Q. Fan and X. Zhong, "A triangle voting algorithm based on double feature constraints for star sensors," *Adv. Space Res.*, vol. 61, no. 4, pp. 1132–1142, Feb. 2018.

[22] J. Li, G. Wang, and X. Wei, "Generation of guide star catalog for star trackers," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4592–4601, Jun. 2018.

[23] H. Kawano, H. Shimoji, and S. Yoshikawa, "Optical testing of star sensor (I): Defocus spot measuring technique for ground-based test," *Opt. Rev.*, vol. 15, no. 2, pp. 110–117, Apr. 2008.

[24] D. D. Needleman, J. P. Alstad, P. C. Lai, and H. M. Elmasri, "Fast access and low memory star pair catalog for star pattern identification," *J. Guid., Control, Dyn.*, vol. 33, no. 5, pp. 1396–1403, Sep. 2010.

[25] E. A. Fox, L. S. Heath, and Q. F. Chen, "Practical minimal perfect hash functions for large databases," *Commun. ACM*, vol. 35, no. 1, pp. 105–121, Jan. 1992.



JIAN LI received the bachelor's and Ph.D. degrees in precision instrument from Beihang University, Beijing, China, from 2004 to 2015. He is currently holding a postdoctoral position at the School of Automation Science and Electrical Engineering, Beihang University. His research interests include the attitude determination sensing systems and image processing.

XINGUO WEI was born in 1977. He received the bachelor's, master's, and Ph.D. degrees from the School of Automation Science and Electrical Engineering, Beihang University, China, from 1995 to 2004. He is currently a Professor with the School of Instrumentation Science and Opto-Electronics Engineering, Beihang University. He has authored over 40 articles, and over ten inventions. His research interests include precision measurement, machine vision, and image processing.

GANGYI WANG received the bachelor's degree in electronics and information engineering, the master's and Ph.D. degrees in information and communication engineering from the Harbin Institute of Technology, in 2006, 2008, and 2013, respectively. He is currently a Lecturer with the School of Instrumentation Science and Opto-Electronics Engineering, Beihang University, China. His research interests include star tracker, computer vision, and embedded systems.

SHUTIAN ZHOU is currently pursuing the B.S. degree in electronic engineering with the University of Electronic Science and Technology of China (UESTC), China. His research interest includes math and information processing.